REMARKS

The Examiner's attention is directed to copies of prior art documents located in the parent file 08/647,769. Since such copies are already available in the parent file, duplicate copies have not been included in the attached PTO-1449.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned "**Version With Markings To Show Changes Made**."

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: Larry S. Nixon

Larry S. Nixon
Reg. No. 25,640

LSN:vsc
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

The paragraph beginning at page 2, line 9:

A further problem with the dissemination of data is that many standards and conventions have developed over the years [therefore]. Therefore, traditionally, users have needed to develop techniques for accessing different data sources, which in itself has tended to become a specialist activity.

The paragraph beginning at page 6, line 3:

Both browsers and servers represent applications which must be made available when requested although, during substantially quiet periods of operation, [there] they are not actually required as such. Procedures of this type, embedded within systems and substantially transparent to operators, have become known, collectively, as "daemons". Thus, in operation, server daemons intermittently interrogate selected ports for incoming signal requests. In response to these requests, servers may make appropriate housekeeping and security operations, whereafter, if possible, the selected file is identified and supplied to the requesting browser.

The paragraph beginning at page 8, line 26:

According to a first aspect of the present invention, there is provided apparatus for serving output signals from a serving device to a plurality of browsing devices connected to a network, wherein said output signals represent commands executable by each browsing device so as to display viewable data in accordance with the specified page format, comprising: means for identifying requests from browsing clients that define a request for specified viewable data with commands for displaying said data in a specified format; means for reading data representing said viewable data; [Means] means for processing said read data so as to combine a representation of said viewable data with executable instructions; and means for supplying output signals to the requesting browsing device derived from said processed data.

The paragraph beginning at page 12, line 24:

Serving station 212, as shown in Figure [1] 2, serves files, processed in accordance with the established hypertext mark-up language (HTML) to browsing clients via the Internet. A browsing client makes a request for the information to be supplied and this request is identified by a serving station, such as station 212, which responds to said request by returning the information via the Internet connection to the browsing client device. Once a request has been received, first signals are processed by the serving station which represent the human viewable data. Second signals are received which represent a selected display structure. These two signals are processed in order to produce an HTML output. However, this processing step only takes place after the client

request has been received such that the first signals and the second signals are processed to produce output signals in the form of client executable instructions which are then served as output signals suitable for execution by the requesting browser. In this way, many pages of HTML encoded data may be produced automatically without requiring manual effort for each individual page. Furthermore, pages may be tailored for specific user requirements and, in some circumstances, it may be possible to adjust the extent to which this [customisation] customization takes place in response to the clients own history of use, such that topics of interest are identified automatically and this identification is used in order to direct information of interest to the calling client.

The paragraph beginning at page 15, line 16:

The HTTP daemon procedures identified at 401 in Figure 4 are detailed in Figure 5. Initialization procedures are implemented at step 501 on start up, whereafter the appropriate port is interrogated at step 503 after waiting for a predetermined period at step 502. The procedures shown in Figure 5 are executed within a multi-tasking environment, therefore the wait period at step 502 refers to a single task and other tasks will execute without being affected. At step 504 a question is asked as to whether a user request, in the form of [an] a uniform resource location (URL) is waiting at the interrogated port. If the question asked at step 504 is answered in the negative, control is returned to step 502 and the process repeated. Thus, as previously stated, the system operates within a multi-tasking environment, such as that provided by the UNIX

operating system. Thus, while the particular tasks shown in Figure 5 repeatedly loop until a URL is received, the system is arranged to perform other tasks.

The paragraph beginning at page 16, line 19:

At step 511 a question is asked as to whether an instruction has been supplied to the effect that CGI.BIN <u>programs</u> are to be executed. If this question is answered in the affirmative, control is directed to step 512, resulting in the execution of the identified CGI.BIN instructions. Alternatively, if the question asked at step 511 is answered in the negative, all possibilities will have been considered and an error message is returned at step 514.